Deterministic Public-Key Encryption for Adaptively Chosen Plaintext Distributions

Deterministic Public-Key Encryption for Adaptively Chosen Plaintext Distributions

Ananth Raghunathan[1⋆], Gil Segev[1⋆⋆], and Salil Vadhan[2]

[1] Computer Science Department
Stanford University, Stanford, CA 94305, USA.
{ananthr,segev}@stanford.edu
[2] School of Engineering and Applied Sciences
& Center for Research on Computation and Society,
Harvard University, Cambridge, MA 02138, USA.
salil@seas.harvard.edu

Abstract. Bellare, Boldyreva, and O'Neill (CRYPTO '07) initiated the study of deterministic public-key encryption as an alternative in scenarios where randomized encryption has inherent drawbacks. The resulting line of research has so far guaranteed security only for adversarially-chosen plaintext distributions that are *independent* of the public key used by the scheme. In most scenarios, however, it is typically not realistic to assume that adversaries do not take the public key into account when attacking a scheme.

We show that it is possible to guarantee meaningful security even for plaintext distributions that depend on the public key. We extend the previously proposed notions of security, allowing adversaries to *adaptively* choose plaintext distributions *after* seeing the public key, in an *interactive* manner. The only restrictions we make are that: (1) plaintext distributions are unpredictable (as is essential in deterministic public-key encryption), and (2) the number of plaintext distributions from which each adversary is allowed to adaptively choose is upper bounded by 2^p, where p can be any predetermined polynomial in the security parameter. For example, with $p = 0$ we capture plaintext distributions that are independent of the public key, and with $p = O(s \log s)$ we capture, in particular, all plaintext distributions that are samplable by circuits of size s.

Within our framework we present both constructions in the random-oracle model based on any public-key encryption scheme, and constructions in the standard model based on lossy trapdoor functions (thus, based on a variety of number-theoretic assumptions). Previously known constructions heavily relied on the independence between the plaintext distributions and the public key for the purposes of randomness extraction. In our setting, however, randomness extraction becomes significantly more challenging once the plaintext distributions and the public key are no longer independent. Our approach is inspired by research on

⋆ Part of the work was done at Microsoft Research Silicon Valley.

⋆⋆ Part of the work was done while the author was a Postdoctoral Researcher at Microsoft Research Silicon Valley.

randomness extraction from seed-dependent distributions. Underlying our approach is a new generalization of a method for such randomness extraction, originally introduced by Trevisan and Vadhan (FOCS '00) and Dodis (PhD Thesis, MIT, '00).

1 Introduction

Deterministic public-key encryption was introduced by Bellare, Boldyreva, and O'Neill [1] as an alternative in scenarios where randomized encryption has inherent drawbacks. For example, ciphertexts that are produced by a randomized encryption algorithm are not length preserving (i.e., may be longer than their corresponding plaintexts), and are in general not efficient searchable – two properties that are problematic in many applications involving massive amounts of data. In addition, the security guarantees provided by randomized public-key encryption schemes are typically highly dependent on the assumption that fresh and essentially uniform random bits are available – which may not always be a valid assumption.

When using a deterministic encryption algorithm, however, the full-fledged notion of semantic security [13] is out of reach. In this light, Bellare et al. initiated the study of formalizing other strong and meaningful notions of security for deterministic public-key encryption, and quite a significant amount of work has been devoted to proposing various such notions and constructing schemes satisfying them [1, 3, 4, 2, 7, 12, 15, 21]. Aiming to obtain as-strong-as-possible notions of security, this recent line of research has successfully shown that a natural variant of the notion of semantic security can be guaranteed even when using a deterministic encryption algorithm, as long as plaintexts are: (1) somewhat *unpredictable*, and (2) *independent* of the public key used by the scheme.

Plaintext unpredictability. When using a deterministic encryption algorithm, essentially no meaningful notion of security can be satisfied when plaintexts are distributed over a small (e.g. polynomial-sized) set. In such a case, an adversary who is given a public key pk and an encryption c of some plaintext m under the public key pk can simply encrypt all possible plaintexts,[3] compare each of them to the given ciphertext c, and thus recover the plaintext m. Therefore, when formalizing a notion of security for deterministic public-key encryption, it is indeed essential to focus on security for unpredictable plaintext distributions.[4]

Key-independent plaintext distributions. Even when dealing with highly unpredictable plaintext distributions, some restrictions should be made on their relation to the public key. Consider, for example, the uniform distribution over plaintexts m subject to the restriction that the first bit of m and the first bit of

[3] More generally, an adversary can encrypt all plaintexts that occurs with at least some non-negligible probability.

[4] Unpredictable plaintext distributions do occur in some natural roles. A prime example is when using a public-key encryption scheme as a key-encapsulation mechanism that encrypts a uniformly distributed key k for a symmetric-key primitive.

$c = \mathsf{Enc}_{pk}(m)$ are equal.[5] More generally, by constructing plaintext distributions that depends on the public key, adversaries can use any *deterministic* encryption algorithm as a *subliminal channel* that leaks much more information on the plaintexts than what any meaningful notion of security should allow.

This paper. For preventing adversaries from exploiting deterministic encryption algorithms as subliminal channels, research on deterministic public-key encryption has so far guaranteed security only for plaintexts distributions that are independent of the public key used by the scheme (which is not realistic, as an adversary can often influence the plaintext distribution after seeing the public key). In this paper, we ask whether or not this is essential. Namely, is it possible to formalize a meaningful notion of security that allows dependencies between plaintext distributions and keys.

1.1 Our Contributions

In this paper, we show that it is *not* essential to focus only on plaintexts distributions that are independent of the keys used by the scheme. We formalize and realize a new notion of security for deterministic public-key encryption, allowing adversaries to *adaptively* choose plaintext distributions *after* seeing the public key of the scheme, in an *interactive* manner. The only restriction we make is that the number of plaintext distributions from which each adversary is allowed to adaptively choose is upper bounded by $2^{p(\lambda)}$, where $p(\lambda)$ can be any predetermined polynomial in the security parameter λ. We stress that the set of $2^{p(\lambda)}$ plaintext distributions can be different for each adversary. Intuitively, this bound says that the *entire* plaintext distribution (not just a single sample) contains at most $p(\lambda)$ bits of information about the public key. We view this as a natural first model for adaptively chosen plaintext distributions, particularly in light of the impossibility of handling arbitrary dependencies (as sketched earlier), and hope that it will pave the way for more realistic models.

Our approach is a generalization of the security notions that have been proposed so far. For example, with $p(\lambda) \equiv 0$ we obtain the notion of security introduced by Bellare, Boldyreva, and O'Neill [1], where the plaintext distribution chosen by the adversary is independent of the public key. As an additional example, with $p(\lambda) = O(s(\lambda) \log s(\lambda))$ we capture, in particular, all plaintext distributions that are samplable by boolean circuits of size at most $s(\lambda)$.

Within our framework we present both generic constructions in the random-oracle model based on any public-key encryption scheme, and generic constructions in the standard model based on lossy trapdoor functions. Our constructions are inspired by the constructions of Bellare, Boldyreva, and O'Neill [1] and of Boldyreva, Fehr, and O'Neill [4]. These constructions rely on the independence between the plaintext distributions and the keys for the purposes of extracting randomness from the plaintext distributions. Randomness extraction becomes significantly more difficult once the plaintexts distributions and the

[5] Note that the support of this distribution will contain nearly half of all plaintexts with high probability.

public keys are no longer independent. Challenges along somewhat similar lines arise in the context of deterministic randomness extraction, where one would like to construct seedless randomness extractors, or seeded randomness extractors for seed-dependent distributions. Indeed, underlying our approach is a new generalization of a method for deterministic extraction, originally introduced by Trevisan and Vadhan [18] and Dodis [9].

Finally, our approach naturally extends to the setting of "hedged" public-key encryption schemes, introduced by Bellare et al. [2]. In this setting, one would like to construct randomized schemes that are semantically secure in the standard sense, and maintain a meaningful and realistic notion of security even when "corrupt" randomness is used by the encryption algorithm. Our notions of adaptive security for deterministic public-key encryption give rise to analogous notions for hedged public-key encryption, and our constructions (when used within the framework of Bellare et al. [2][6]) yield the first adaptively-secure hedged public-key encryption schemes.

1.2 Related Work

The formal study of deterministic public-key encryption was initiated by Bellare, Boldyreva, and O'Neill [1], following research on symmetric-key encryption of high-entropy messages by Russell and Wang [17] and Dodis and Smith [10]. Bellare et al. formalized several notions of security, which were later refined and extended by Bellare, Fischlin, O'Neill, and Ristenpart [3], and by Boldyreva, Fehr, and O'Neill [4]. Bellare, Boldyreva, and O'Neill presented constructions in the random oracle model, and constructions in the standard model were first presented by Bellare, Boldyreva, and O'Neill, and additionally by Boldyreva, Fehr, and O'Neill. Brakerski and Segev [7] showed that the min-entropy requirement considered in all previous works on deterministic public-key encryption can be relaxed to consider hard-to-invert auxiliary inputs. Based on specific number-theoretic assumptions, they designed schemes that are secure in the more general auxiliary-input model, and their constructions were later unified by Wee [21]. Progress along similar lines was made by Fuller, O'Neill and Reyzin [12], who presented a scheme that can securely encrypt a small predetermined number of plaintexts with arbitrary dependencies as long as each has high min-entropy. Additional progress in studying deterministic public-key encryption schemes was recently made by Mironov, Pandey, Reingold, and Segev [15] who constructed such schemes with optimal incrementality.

A step towards obtaining adaptive security for deterministic public-key encryption was made by Bellare et al. [2] who defined and constructed "hedged" public-key encryption schemes (discussed in Section 1.1). Whereas the notions of security considered in [1, 3, 4, 7, 21, 12, 15] capture only "single-shot" adversaries (i.e., adversaries that challenge the given scheme with only one plaintext distribution), Bellare et al. [2] showed that it is possible to guarantee security even

[6] For example, as part of their generic "pad-then-deterministic" scheme, which deterministically encrypts the concatenation of the plaintext and the randomness.

against "multi-shot" adversaries (i.e., adversaries that interactively challenge the scheme with plaintext distributions depending on previous ciphertexts that they received). In their notion of security, however, adversaries are not given access to the public key that is being attacked. In our work we consider the more general, and more typical, scenario where adversaries are given *direct access* to the public key being attacked (and are allowed to adaptively and interactively choose plaintext distributions depending on previous ciphertexts that they received).[7] As discussed in Section 1.1, our constructions yield the first adaptively-secure hedged public-key encryption schemes.

1.3 Overview of Our Approach

In this section we provide a high-level overview of our notions of security and of the main ideas underlying our constructions. We focus here on our constructions in the standard model (i.e., without random oracles), as these emphasize more clearly the main challenges in designing encryption schemes satisfying our notions of security.

Our notions of security. As discussed above, our notions of security for deterministic public-key encryption differ from the previously proposed ones by providing adversaries with *direct* access to the public key. Specifically, we formalize security via a game between an adversary and a "real-or-random" encryption oracle. First, a pair containing a public key and a secret key is produced using the key-generation algorithm of the scheme under consideration, and the adversary is given the public key. Then, the adversary adaptively interacts with the encryption oracle, where each query consists of a description of a plaintext distribution M. For simplicity, here we consider distributions over plaintexts, but in fact our notion allows distributions over blocks of plaintexts. The encryption oracle operates in one of two modes, "real" or "random", which is chosen uniformly at random at the beginning of the game. In the "real" mode, the encryption oracle samples a plaintext according to M, and the adversary is given its encryption under the public key. In the "random" mode, the encryption oracle samples a plaintext from the uniform distribution over the plaintext space, and the adversary is again given its encryption under the public key.[8]

The goal of the adversary in this game is to distinguish between the "real" mode and "random" mode with a non-negligible probability, subject only to the

[7] In fact, the approach of Bellare et al. [2] relies on encryption schemes in which ciphertexts reveal essentially no information on the corresponding public key. Therefore, even multi-shot adversaries learn essentially no information on the public key being attacked, and thus their "adaptive" choices of plaintext distributions are still independent of the public key. This approach does not seem to extend to our setting, where adversaries are given direct access to the public key.

[8] We note that the resulting notion of security is polynomially equivalent (via a standard hybrid argument) to an analogous "left" or "right" formulation in which the adversary specifies two plaintext distributions, and the encryption oracle uses either the left one of the right one.

requirement that for any such adversary there exists a set $\mathcal{X} = \mathcal{X}_\lambda$ of plaintext distributions such that:

1. $|\mathcal{X}| \leq 2^p$, where $p = p(\lambda)$ is any predetermined polynomial in the security parameter (the construction of the scheme can depend on the polynomial p).
2. The adversary queries the encryption oracle only with plaintext distributions in $\mathcal{X}$.
3. Each plaintext distribution in $\mathcal{X}$ has min-entropy at least k, where $k = k(\lambda)$ is a predetermined function of the security parameter.

In addition, we naturally extend the above game to capture chosen-ciphertext attacks, by allowing adversaries adaptive access to a decryption oracle (subject to the standard requirement of not querying the decryption oracle with any ciphertext that was produced by the encryption oracle).

We note that our security game is in fact almost identical to the standard "real-or-random" one for randomized public-key encryption. Specifically, unlike the previously proposed notions of security for deterministic public-key encryption, we provide the adversary with direct access to the public key, and allow the adversary to adaptively interact with the encryption and decryption oracles *in any order*.[9]

Chosen-plaintext security in the standard model. The starting point for our construction is the one of Boldyreva, Fehr, and O'Neill, which we now briefly describe. In their construction, the public key consists of a function f that is sampled from the injective mode of a collection of lossy trapdoor functions, and a permutation π sampled from a pairwise-independent collection of permutations. (We refer the reader to Section 2 for the relevant definitions.) The secret key consists of the trapdoor for inverting f. (We require that π is efficiently invertible.) The encryption of a message m is defined as $\mathsf{Enc}_{pk}(m) = f(\pi(m))$, and decryption is naturally defined.

The proof of security consists of two steps. First, the security of the collection of lossy trapdoor functions allows one to replace the injective function f with a lossy function $\widetilde{f}$ (where lossy means that the size of $\widetilde{f}$'s image is significantly smaller than the size of its domain). Then, the Crooked Leftover Hash Lemma of Dodis and Smith [10] states that for any plaintext distribution M that has a certain amount of min-entropy, for a uniformly and independently chosen pairwise-independent permutation π it holds that the distributions $\widetilde{f}(\pi(M))$ and $\widetilde{f}(U)$ are statistically close (even given $\widetilde{f}$ and π), where U is the uniform distribution over plaintexts. That is, essentially no information on the plaintext is revealed.

This construction, however, becomes insecure when adversaries can choose the plaintext distribution M after receiving the description of π. Specifically, the Crooked Leftover Hash Lemma no longer holds when M may depend on π, and

[9] In contrast, due to requiring key-independent plaintext distributions, Bellare et al. [1] and Boldyreva et al. [4] allow chosen-ciphertext adversaries to query the decryption oracle *only after* they have queried the encryption oracle.

adversaries may easily use the encryption algorithm as a subliminal channel for leaking information about the plaintext, as discussed above.

The main idea underlying our basic construction is to sample the permutation π from a collection of highly-independent permutations. We prove that this modification results in a scheme that is secure according to our new notion of security by proving a *High-Moment* Crooked Leftover Hash Lemma.[10] Informally, we prove that for any lossy function $\widetilde{f}$, and for any set $\mathcal{X}$ of sources with a certain amount of min-entropy, with an overwhelming probability over the choice of a permutation π from a t-wise almost-independent collection of permutations (where t depends only logarithmically on the size of $\mathcal{X}$), for *every* $M \in \mathcal{X}$ it holds that $\widetilde{f}(\pi(M))$ and $\widetilde{f}(U)$ are statistically close. In particular, in such a setting the specific choice of $M \in \mathcal{X}$ can adaptively depend on the permutation π, and still the statistical distance is negligible.

Chosen-ciphertext security in the standard model. While in the setting of chosen-plaintext security our construction is a natural generalization of that of Boldyreva et al. [4] (given our high-moment generalization of the crooked leftover hash), this is not the case in the setting of chosen-ciphertext security. In this setting, the CCA-secure scheme of Boldyreva et al. relies more strongly on the assumption that the challenge plaintext distribution is independent of the public key of the scheme (not just in the context of the Crooked Leftover Hash Lemma as above) – an assumption that we do not make. Nevertheless, we show that some of the ideas underlying their approach can still be utilized to construct a scheme that is secure according to our notion of security.

The scheme of Boldyreva et al. follows the "all-but-one" simulation paradigm of Peikert and Waters [16] using all-but-one lossy trapdoor functions. These are tag-based functions, where one of the tags corresponds to a lossy function, and all other tags correspond to injective functions. As in the work of Peikert and Waters [16], the approach of Boldyreva et al. makes sure that the challenge plaintext corresponds to a lossy tag (and thus the challenge ciphertext reveals no information), while all other plaintexts corresponds to injective tags (and a suitable simulator is able to properly simulate the decryption oracle). When dealing with a deterministic encryption algorithm, note that tags must be derived deterministically from the plaintext and the public key. The approach of Boldyreva et al. is based on first sampling the challenge plaintext m^*, and only then generating a public key for which m^* corresponds to a lossy tag, but all other plaintexts correspond to injective tags.

This approach fails in our setting, where adversaries specify the distribution of the challenge plaintext in an adaptive manner as a function of the public

[10] As already noted, a high-moment generalization of the (standard) Leftover Hash Lemma was given by Trevisan and Vadhan [18] and Dodis [9], but no analogous generalization was known for the crooked leftover hash lemma. A different high-moment generalization of the crooked leftover hash lemma was proved by Fuller et al. [12] for the purpose of extracting randomness from a small number of possibly correlated sources. However, their generalization does not allow seed-dependent sources, and therefore allows only non-adaptive adversaries.

key. Thus, in our setting we must be able to generate a public key before the challenge plaintext is known. We note that a somewhat similar issue arises in the setting of identity-based encryption (IBE): "selective security" considers adversaries that specify the challenge identity in advance, whereas "full security" considers adversaries that can adaptively choose the challenge identity. One simple solution that was proposed in the IBE setting is to a-priori guess the challenge identity, and this solution naturally extends to our setting by guessing the tag corresponds to the challenge plaintext. This, however, requires sub-exponential hardness assumptions, which we aim to avoid.

Our approach is based on the one of Boneh and Boyen [5] (and on its refinement by Cash, Hofheinz, Kiltz, and Peikert [8] for converting a large class of selectively-secure IBE schemes to fully-secure ones,[11] combined with the idea of $\mathcal{R}$-lossiness due to Boyle, Segev, and Wichs [6]. Specifically, we derive tags from plaintexts using an admissible hash functions [5, 8], and instead of using all-but-one lossy trapdoor functions, we introduce the notion of $\mathcal{R}$-lossy trapdoor functions (which we generically construct based on lossy trapdoor functions).[12] This is a generalization of the notion of all-but-one lossy trapdoor functions, where the set of tags is partitioned into lossy tags and injective tags according to the relation $\mathcal{R}$. (In particular, there may be more than one lossy tag.) Combined with an admissible hash function, we are able to ensure that even with an adaptive adversary, with some non-negligible probability, the challenge plaintext corresponds to a lossy tag (and thus the challenge ciphertext reveals no information), while all other plaintexts corresponds to injective tags (and a suitable simulator is able to properly simulate the decryption oracle). We show that such a guarantee enables us to prove the security of our scheme with respect to adaptive adversaries.

1.4 Paper Organization

GIL: To be written.

2 Preliminaries

GIL: Need to filter out anything that will not be used (probably Lemma 2.1 and subsection on admissible hash functions).

For an integer $n \in \mathbb{N}$ we denote by $[n]$ the set $\{1, \ldots, n\}$, and by U_n the uniform distribution over the set $\{0,1\}^n$. For a random variable X we denote by $x \leftarrow X$ the process of sampling a value x according to the distribution of X and by $\mathbb{E}[X]$ the expectation of the random variable X. Similarly, for a finite set S

[11] We note that the work of Cash et al. [8] is based on ideas introduced by Boneh and Boyen [5] and Waters [20].

[12] Boyle, Segev and Wichs [6] introduced the notion of $\mathcal{R}$-lossy public-key encryption, which can be viewed as a randomized variant of our notion of $\mathcal{R}$-lossy trapdoor functions.

we denote by $x \leftarrow S$ the process of sampling a value x according to the uniform distribution over S. We denote by $\boldsymbol{X} = (X_1, \ldots, X_T)$ a joint distribution of T random variables, and by $\boldsymbol{x} = (x_1, \ldots, x_T)$ a sample drawn from $\boldsymbol{X}$. For two bit-strings x and y we denote by $x \| y$ their concatenation. A non-negative function $f : \mathbb{N} \to \mathbb{R}$ is *negligible* if it vanishes faster than any inverse polynomial.

In this paper we consider the uniform adversarial model (i.e. consider uniform probabilistic polynomial-time adversaries). We note that all of our results also apply to the nonuniform adversarial model (under nonuniform complexity assumptions).

The *min-entropy* of a random variable X is $\mathbf{H}_\infty(X) = -\log(\max_x \Pr[X = x])$. A *$k$-source* is a random variable X with $\mathbf{H}_\infty(X) \geq k$. A *$(T, k)$-source* is a random variable $\boldsymbol{X} = (X_1, \ldots, X_T)$ where each X_i is a k-source for every $i \in [T]$. A *(T, k)-block source* is a random variable $\boldsymbol{X} = (X_1, \ldots, X_T)$ where for every $i \in [T]$ and $x_1, \ldots, x_{i-1}$ it holds that $\mathbf{H}_\infty(X_i | X_1 = x_1, \ldots, X_{i-1} = x_{i-1}) \geq k$.

The following standard lemma states that conditioning on random variable that obtains at most 2^v values can reduce the min-entropy of any other random variable by essentially at most v.

Lemma 2.1 (cf. [19, Lemma 6.30]). *Let (Z, X) be any two jointly distributed random variables such that $|\mathrm{Supp}(Z)| \leq 2^v$. Then, for any $\epsilon > 0$ it holds that*

$$\Pr_{z \leftarrow Z}[\mathbf{H}_\infty(X | Z = z) \geq \mathbf{H}_\infty(X) - v - \log(1/\epsilon)] \geq 1 - \epsilon.$$

The *statistical distance* between two random variables X and Y over a finite domain Ω is $\mathbf{SD}(X, Y) = \frac{1}{2} \sum_{\omega \in \Omega} |\Pr[X = \omega] - \Pr[Y = \omega]|$. Two random variables X and Y are *δ-close* if $\mathbf{SD}(X, Y) \leq \delta$. Two distribution ensembles $\{X_\lambda\}_{\lambda \in \mathbb{N}}$ and $\{Y_\lambda\}_{\lambda \in \mathbb{N}}$ are *statistically indistinguishable* if it holds that $\mathbf{SD}(X_\lambda, Y_\lambda)$ is negligible in λ. They are *computationally indistinguishable* if for every probabilistic polynomial-time algorithm $\mathcal{A}$ it holds that

$$\left| \Pr_{x \leftarrow X_\lambda}\left[\mathcal{A}(1^\lambda, x) = 1\right] - \Pr_{y \leftarrow Y_\lambda}\left[\mathcal{A}(1^\lambda, y) = 1\right] \right|$$

is negligible in λ.

2.1 t-Wise δ-Dependent Permutations

A collection Π of permutations over $\{0,1\}^n$ is *t-wise δ-dependent* if for any distinct $x_1, \ldots, x_t \in \{0,1\}^n$ the distribution $(\pi(x_1), \ldots, \pi(x_t))$ where π is sampled from Π is δ-close in statistical distance to the distribution $(\pi^*(x_1), \ldots, \pi^*(x_t))$ where π^* is a truly random permutation. For our construction in the standard model we rely on an explicit construction of such a collection due to Kaplan, Naor, and Reingold [14] that enjoys an asymptotically optimal description length (although we note that in fact any other construction can be used):

Theorem 2.2 ([14]). *For any integers n and $t \leq 2^n$, and for any $0 < \delta < 1$, there exists an explicit t-wise δ-dependent collection Π of permutations over $\{0,1\}^n$ where each permutation $\pi \in \Pi$ can be described using $O(nt + \log(1/\delta))$ bits, and is computable and invertible in time polynomial in n, t and $\log(1/\delta)$.*

2.2 Admissible Hash Functions

The concept of an *admissible hash function* was first defined by Boneh and Boyen [5] to convert a large class of selectively-secure identity-based encryption scheme into a fully-secure ones. In this paper we use such hash functions in a somewhat similar way as part of our construction of a CCA-secure deterministic public-key encryption scheme. The main idea of an admissible hash function is that it allows the reduction in the proof of security to secretly partition the message space into two subsets, which we will label as "lossy tags" and "injective tags," such that there is a noticeable probability that all of the messages in the adversary's decryption queries will correspond to injective tags, but the challenge ciphertext will correspond to a lossy tag. This is useful if the simulator can efficiently answer decryption queries with injective tags, while a challenge ciphertext with a lossy tag reveals essentially no information on the encrypted message. Our exposition and definition of admissible hash function follows that of Cash, Hofheinz, Kiltz, and Peikert [8].

For $K \in \{0,1,\perp\}^{v(\lambda)}$, we define the "partitioning" function $P_K : \{0,1\}^{v(\lambda)} \to \{\texttt{Lossy}, \texttt{Inj}\}$ which partitions the space $\{0,1\}^{v(\lambda)}$ of tags in the following way:

$$P_K(y) := \begin{cases} \texttt{Lossy} & \text{if } \forall\, i \in \{1,\ldots,v(\lambda)\} \quad : \quad K_i = y_i \text{ or } K_i = \perp \\ \texttt{Inj} & \text{otherwise} \end{cases}$$

For any $u = u(\lambda) < v(\lambda)$, we let $\mathcal{K}_{u,\lambda}$ denote the uniform distribution over $\{0,1,\perp\}^{v(\lambda)}$ *conditioned on* exactly u positions having $\perp$ values. (Note, if K is chosen from $\mathcal{K}_{u,\lambda}$, then the map $P_K(\cdot)$ defines exactly 2^u values as $\texttt{Lossy}$.) We would like to pick a distribution $\mathcal{K}_{u,\lambda}$ for choosing K so that, there is a noticeable probability for every set of tags $y_0,\ldots,y_q$, of y_0 being classified as "lossy" and all other tags "injective." Unfortunately, this cannot happen if we allow all tags. Instead, we will need to rely on a special hash function the maps messages x to tags y.

Definition 2.3 (Admissible hash functions [5, 8]). *Let $\mathcal{H} = \{\mathcal{H}_\lambda\}_{\lambda\in\mathbb{N}}$ be a hash-function ensemble, where each $h \in \mathcal{H}_\lambda$ is a polynomial-time computable function $h : \{0,1\}^{n(\lambda)} \to \{0,1\}^{v(\lambda)}$. We say that $\mathcal{H}$ is an* admissible hash-function ensemble *if for every $h \in \mathcal{H}$ there exists a efficiently recognizable set $\mathsf{Unlikely}_h \subseteq \bigcup_{q\in\mathbb{N}} \left(\{0,1\}^{n(\lambda)}\right)^q$ of string-tuples such that the following two properties hold:*

- *For every probabilistic polynomial-time algorithm $\mathcal{A}$ there exists a negligible function $\nu(\lambda)$ satisfying*

$$\Pr[(x_0,\ldots,x_q) \in \mathsf{Unlikely}_h] \leq \nu(\lambda),$$

  where $h \leftarrow \mathcal{H}_\lambda$ and $(x_0,\ldots,x_q) \leftarrow \mathcal{A}(1^\lambda, h)$.
- *For every polynomial $q = q(\lambda)$ there is a polynomial $\Delta = \Delta(\lambda)$ and an efficiently computable $u = u(\lambda)$ such that, for every $h \in \mathcal{H}_\lambda$ and $(x_0,\ldots,x_q) \notin \mathsf{Unlikely}_h$ with $x_0 \notin \{x_1,\ldots,x_q\}$ we have:*

$$\Pr_{K\leftarrow\mathcal{K}_{u,\lambda}}[P_K(h(x_0)) = \texttt{Lossy} \wedge P_K(h(x_1)) = \cdots = P_K(h(x_q)) = \texttt{Inj}\ ] \geq \frac{1}{\Delta(\lambda)}.$$

The work of Boneh and Boyen [5] shows how to construct admissible hash functions from collision-resistant hash functions.

2.3 Lossy Trapdoor Functions

A collection of lossy trapdoor functions [16] consists of two families of functions. Functions in one family are injective and can be efficiently inverted using a trapdoor. Functions in the other family are "lossy," which means that the size of their image is significantly smaller than the size of their domain. The only security requirement is that a description of a randomly chosen function from the family of injective functions is computationally indistinguishable from a description of a randomly chosen function from the family of lossy functions.

Definition 2.4 (Lossy trapdoor functions [16, 11]). *Let $n : \mathbb{N} \to \mathbb{N}$ and $\ell : \mathbb{N} \to \mathbb{N}$ be non-negative functions, and for any $\lambda \in \mathbb{N}$ let $n = n(\lambda)$ and $\ell = \ell(\lambda)$. A* collection of (n, ℓ)-lossy trapdoor functions *is a 4-tuple of probabilistic polynomial-time algorithms $(\mathsf{Gen}_0, \mathsf{Gen}_1, \mathsf{F}, \mathsf{F}^{-1})$ such that:*

1. **Sampling a lossy function:** $\mathsf{Gen}_0(1^\lambda)$ *outputs a function index $\sigma \in \{0,1\}^*$.*
2. **Sampling an injective function:** $\mathsf{Gen}_1(1^\lambda)$ *outputs a pair $(\sigma, \tau) \in \{0,1\}^* \times \{0,1\}^*$, where σ is a function index and τ is a trapdoor.*
3. **Evaluation:** *Let $n = n(\lambda)$ and $\ell = \ell(\lambda)$. Then, for every function index σ produced by either Gen_0 or Gen_1, the algorithm $\mathsf{F}(\sigma, \cdot)$ computes a function $f_\sigma : \{0,1\}^n \to \{0,1\}^*$ with one of the two following properties:*
   - *Lossy: If σ is produced by Gen_0, then the image of f_σ has size at most $2^{n-\ell}$.*
   - *Injective: If σ is produced by Gen_1, then the function f_σ is injective.*
4. **Inversion of injective functions:** *For every pair (σ, τ) produced by Gen_1 and every $x \in \{0,1\}^n$, we have $\mathsf{F}^{-1}(\tau, \mathsf{F}(\sigma, x)) = x$.*
5. **Security:** *The two ensembles $\{\sigma : \sigma \leftarrow \mathsf{Gen}_0(1^\lambda)\}_{\lambda \in \mathbb{N}}$ and $\{\sigma : (\sigma, \tau) \leftarrow \mathsf{Gen}_1(1^\lambda)\}_{\lambda \in \mathbb{N}}$ are computationally indistinguishable.*

Constructions of lossy trapdoor functions were proposed based on a wide variety of number-theoretic assumptions and for a large range of parameters (see, for example, [11, 16] and the references therein). In particular, in terms of parameters, several constructions are known to offer $\ell = n - n^\epsilon$ for any fixed constant $0 < \epsilon < 1$ with $n = \mathrm{poly}(\lambda)$.

2.4 Deterministic Public-Key Encryption

A deterministic public-key encryption scheme is a triplet $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ of polynomial-time algorithms with the following properties:

- The key-generation algorithm KeyGen is a randomized algorithm that takes as input the security parameter 1^λ and outputs a key pair (sk, pk) consisting of a secret key sk and a public key pk.

- The encryption algorithm Enc is a *deterministic* algorithm that takes as input a public key pk and a message $m \in \{0,1\}^{n(\lambda)}$, and outputs a ciphertext $c = \mathsf{Enc}_{pk}(m)$.
- The decryption algorithm is a possibly randomized algorithm that takes as input a secret key sk and a ciphertext c and outputs a message $m \leftarrow \mathsf{Dec}_{sk}(c)$ such that $m \in \{0,1\}^{n(\lambda)} \cup \{\perp\}$.

3 Formalizing Adaptive Security for Deterministic Public-Key Encryption

In this section we present a framework for modeling the security of deterministic public-key encryption schemes in an *adaptive* setting. As discussed in Section 1.3, we consider adversaries that *adaptively* choose plaintext distributions *after* seeing the public key of the scheme, in an *interactive* manner. The only restriction we make is that the *number* of plaintext distributions from which each adversary is allowed to choose is upper bounded by $2^{p(\lambda)}$, where $p(\lambda)$ can be any a-priori given polynomial in the security parameter λ. The security definitions that follow are parameterized by three parameters:

- $p = p(\lambda)$ denoting the 2^p bound on the number of allowed plaintext distributions.
- $T = T(\lambda)$ denoting the number of blocks in each plaintext distribution.
- $k = k(\lambda)$ denoting the min-entropy requirement.

Additionally, they are implicitly parameterized by bit-length $n = n(\lambda)$ of plaintexts. We begin by defining the "real-or-random" encryption oracle which we use formalize security.

Definition 3.1 (Real-or-random encryption oracle). *The real-or-random oracle* RoR *takes as input triplets of the form* $(\mathsf{mode}, pk, \boldsymbol{M})$*, where* $\mathsf{mode} \in \{\mathsf{real}, \mathsf{rand}\}$*, pk is a public key, and $\boldsymbol{M} = (M_1, \ldots, M_T)$ is a circuit representing a joint distribution over T messages. If* $\mathsf{mode} = \mathsf{real}$ *then the oracle samples $(m_1, \ldots, m_T) \leftarrow \boldsymbol{M}$, and if* $\mathsf{mode} = \mathsf{rand}$ *then the oracle samples $(m_1, \ldots, m_T) \leftarrow U^T$ where U is the uniform distribution over the appropriate message space. It then outputs the vector of ciphertexts* $(\mathsf{Enc}_{pk}(m_1), \ldots, \mathsf{Enc}_{pk}(m_T))$.

Following [1, 4] we consider two classes of adversarially-chosen message distributions $\boldsymbol{M} = (M_1, \ldots, M_T)$: The class of (T,k)-sources, where each M_i is assumed to be a k-source, and the more restrictive class of (T,k)-block-sources, where each M_i is assumed to be a k-source even given $M_1, \ldots, M_{i-1}$. (See Section 2 for formal definitions.) Our constructions in the random oracle model are secure with respect to (T,k)-sources, and our constructions in the standard model are secure with respect to (T,k)-block-sources. This gap was recently shown by Wichs [22] to be inherent to our techniques, and in fact to all the techniques that were so far used for designing deterministic public-key encryption schemes without random oracles [3, 4, 2, 7, 12, 15, 21]. Specifically, Wichs showed that no

deterministic public-key encryption scheme can be proven secure for all (T, k)-sources using a black-box reduction to a "falsifiable" hardness assumption. (We refer the reader to [22] for more details on his notion of falsifiability.)

The following two definitions capture the class of chosen-plaintext adversaries and security game that we consider in this paper. We refer the reader to the full version [**?**] for their natural generalization to chosen-ciphertext attacks.

Definition 3.2 (2^p-bounded (T, k)-source adversary). *Let $\mathcal{A}$ be a probabilistic polynomial-time algorithm that is given as input a pair $(1^\lambda, pk)$ and oracle access to $\mathsf{RoR}(\mathsf{mode}, pk, \cdot)$ for some $\mathsf{mode} \in \{\mathsf{real}, \mathsf{rand}\}$. Then, $\mathcal{A}$ is a* 2^p-bounded (T, k)-source adversary *if for every $\lambda \in \mathbb{N}$ there exists a set $\mathcal{X} = \mathcal{X}_\lambda$ of polynomial-time samplable (T, k)-sources such that:*

1. *$|\mathcal{X}| \leq 2^p$.*
2. *For each of $\mathcal{A}$'s RoR queries $\boldsymbol{M}$ it holds that:*
   - *$\boldsymbol{M} \in \mathcal{X}$.*
   - *For all $(m_1, \ldots, m_T)$ in the support of $\boldsymbol{M}$ and for all distinct $i, j \in [T]$ it holds that $m_i \neq m_j$.*

In addition, $\mathcal{A}$ is a block-source *adversary if $\mathcal{X}$ is a set of (T, k)-block-sources.*

Definition 3.3 (Adaptive chosen-distribution attacks (ACD-CPA)). *A deterministic public-key encryption scheme $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is (p, T, k)-*ACD-CPA-secure *(resp.* block-wise (p, T, k)-ACD-CPA-secure*) if for any probabilistic polynomial-time 2^p-bounded (T, k)-source (resp. block-source) adversary $\mathcal{A}$, there exists a negligible function $\nu(k)$ such that*

$$\mathbf{Adv}^{\text{ACD-CPA}}_{\Pi,\mathcal{A}}(\lambda) \stackrel{\mathsf{def}}{=} \left|\Pr\left[\mathsf{Expt}^{\mathsf{real}}_{\Pi,\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{rand}}_{\Pi,\mathcal{A}}(\lambda) = 1\right]\right| \leq \nu(\lambda),$$

where for each $\mathsf{mode} \in \{\mathsf{real}, \mathsf{rand}\}$ and $\lambda \in \mathbb{N}$ the experiment $\mathsf{Expt}^{\mathsf{mode}}_{\Pi,\mathcal{A}}(\lambda)$ is defined as follows:

1. *$(sk, pk) \leftarrow \mathsf{KeyGen}(1^\lambda)$.*
2. *$b \leftarrow \mathcal{A}^{\mathsf{RoR}(\mathsf{mode}, pk, \cdot)}(1^\lambda, pk)$.*
3. *Output b.*

In addition, such a scheme is (p, T, k)-ACD1-CPA-secure *(resp.* block-wise (p, T, k)-ACD1-CPA-secure*) if the above holds for any probabilistic polynomial-time 2^p-bounded (T, k)-source (resp. block-source) adversary $\mathcal{A}$ that queries the RoR oracle at most once.*

Our adaptive notion of security enables an immediate reduction of "multi-shot" adversaries to "single-shot" ones, as in the case of randomized public-key encryption. The following theorem follows via a standard hybrid argument.

Theorem 3.4 (Equivalence of ACD-CPA-security and ACD1-CPA-security). *For any p, T, and k, a deterministic public-key encryption scheme Π is (p, T, k)-*ACD-CPA*-secure (resp. block-wise (p, T, k)-*ACD-CPA*-secure) if and only if it is (p, T, k)-*ACD1-CPA*-secure (resp. block-wise (p, T, k)-*ACD1-CPA*-secure).*

4 Chosen-Plaintext Security based on Lossy Trapdoor Functions

In this section we present our basic construction of a public-key deterministic encryption scheme that is secure according to our notion of adaptive security. We refer the reader to Section 1.3 for a high-level description of the scheme, and of the main challenges and ideas underlying our approach. In what follows we formally describe the scheme, discuss the parameters that we obtain using known instantiations of its building blocks, and discuss the main ideas underlying its proof of security.

The scheme $\mathcal{DE}$. Let $n = n(\lambda)$, $\ell = \ell(\lambda)$, $t = t(\lambda)$ and $\delta = \delta(\lambda)$ be functions of the security parameter $\lambda \in \mathbb{N}$. Let $(\mathsf{Gen}_0, \mathsf{Gen}_1, \mathsf{F}, \mathsf{F}^{-1})$ be a collection of (n, ℓ)-lossy trapdoor functions, and for every $\lambda \in \mathbb{N}$ let Π_λ be a t-wise δ-dependent collection of permutations over $\{0,1\}^n$. Our scheme $\mathcal{DE} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is defined as follows:

- **Key generation.** The key-generation algorithm KeyGen on input 1^λ samples $(\sigma, \tau) \leftarrow \mathsf{Gen}_1(1^\lambda)$ and $\pi \leftarrow \Pi_\lambda$. It then outputs $pk = (\sigma, \pi)$ and $sk = \tau$.
- **Encryption.** The encryption algorithm Enc on input a public key $pk = (\sigma, \pi)$ and a message $m \in \{0,1\}^n$ outputs $c = \mathsf{F}(\sigma, \pi(m))$.
- **Decryption.** The decryption algorithm Dec on input a secret key $sk = \tau$ and a ciphertext c outputs $m = \pi^{-1}\left(\mathsf{F}^{-1}(\tau, c)\right)$.

Theorem 4.1. *The scheme $\mathcal{DE}$ is block-wise (p, T, k)-*ACD-CPA*-secure for any $n = n(\lambda)$, $\ell = \ell(\lambda)$, $p = p(\lambda)$, and $T = T(\lambda)$ by setting $t = p + n - \ell + \log T + \omega(\log \lambda)$, $k = n - \ell + 2\log T + 2\log t + \omega(\log \lambda)$, and $\delta = 2^{-nt}$.*

Parameters. Using existing constructions of lossy trapdoor functions (see Section 2.3), for any $n = n(\lambda)$ and for any constant $0 < \epsilon < 1$ we can instantiate our scheme with $\ell = n - n^\epsilon$. Therefore, for any $n = n(\lambda)$, $p = p(\lambda)$, and $T = T(\lambda)$, we obtain schemes with $t = p + n^\epsilon + \omega(\log \lambda)$, $k = n^\epsilon + \omega(\log \lambda)$, and $\delta = 2^{-nt}$.

Proof overview. The proof of security consists of two steps. Let $\mathcal{X}$ be a set of at most 2^p plaintext distributions. First, the security of the collection of lossy trapdoor functions allows us to replace the injective function $f(\cdot) = \mathsf{F}(\sigma, \cdot)$ with a lossy function $\widetilde{f}(\cdot) = \mathsf{F}(\tilde{\sigma}, \cdot)$. Next, we use the high-moment crooked leftover hash lemma derived in Section **??** and show that with overwhelming probability over the choice of the permutation π, it holds that for *every* plaintext distribution $\boldsymbol{M} \in \mathcal{X}$, the two distributions $\widetilde{f}(\pi(\boldsymbol{M}))$ and $\widetilde{f}(\boldsymbol{U})$ are statistically close, even given the public key (i.e., $\tilde{\sigma}$ and π). Therefore, essentially no information on the plaintext is revealed – even when the specific choice of $\boldsymbol{M} \in \mathcal{X}$ may adaptively depend on pk. A second application of the security of the collection of lossy trapdoor functions allows us to switch back from the lossy function to an injective one, which exactly reflects the output of the real-or-random encryption oracle in the rand mode. We refer the reader to the full version [**?**] for the formal proof.

References

1. M. Bellare, A. Boldyreva, and A. O'Neill. Deterministic and efficiently searchable encryption. In *Advances in Cryptology – CRYPTO '07*, pages 535–552, 2007.
2. M. Bellare, Z. Brakerski, M. Naor, T. Ristenpart, G. Segev, H. Shacham, and S. Yilek. Hedged public-key encryption: How to protect against bad randomness. In *Advances in Cryptology – ASIACRYPT '09*, pages 232–249, 2009.
3. M. Bellare, M. Fischlin, A. O'Neill, and T. Ristenpart. Deterministic encryption: Definitional equivalences and constructions without random oracles. In *Advances in Cryptology – CRYPTO '08*, pages 360–378, 2008.
4. A. Boldyreva, S. Fehr, and A. O'Neill. On notions of security for deterministic encryption, and efficient constructions without random oracles. In *Advances in Cryptology – CRYPTO '08*, pages 335–359, 2008.
5. D. Boneh and X. Boyen. Secure identity based encryption without random oracles. In *Advances in Cryptology – CRYPTO '04*, pages 443–459, 2004.
6. E. Boyle, G. Segev, and D. Wichs. Fully leakage-resilient signatures. In *Advances in Cryptology – EUROCRYPT '11*, pages 89–108, 2011.
7. Z. Brakerski and G. Segev. Better security for deterministic public-key encryption: The auxiliary-input setting. In *Advances in Cryptology – CRYPTO '11*, pages 543–560, 2011.
8. D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. In *Advances in Cryptology – EUROCRYPT '10*, pages 523–552, 2010.
9. Y. Dodis. *Exposure-Resilient Cryptography.* PhD thesis, MIT, 2000.
10. Y. Dodis and A. Smith. Entropic security and the encryption of high entropy messages. In *Proceedings of the 2nd Theory of Cryptography Conference*, pages 556–577, 2005.
11. D. M. Freeman, O. Goldreich, E. Kiltz, A. Rosen, and G. Segev. More constructions of lossy and correlation-secure trapdoor functions. *J. Cryptology*, 26(1):39–74, 2013.
12. B. Fuller, A. O'Neill, and L. Reyzin. A unified approach to deterministic encryption: New constructions and a connection to computational entropy. In *Proceedings of the 9th Theory of Cryptography Conference*, pages 582–599, 2012.
13. S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.
14. E. Kaplan, M. Naor, and O. Reingold. Derandomized constructions of k-wise (almost) independent permutations. *Algorithmica*, 55(1):113–133, 2009.
15. I. Mironov, O. Pandey, O. Reingold, and G. Segev. Incremental deterministic public-key encryption. In *Advances in Cryptology – EUROCRYPT '12*, pages 628–644, 2012.
16. C. Peikert and B. Waters. Lossy trapdoor functions and their applications. *SIAM Journal on Computing*, 40(6):1803–1844, 2011.
17. A. Russell and H. Wang. How to fool an unbounded adversary with a short key. *IEEE Transactions on Information Theory*, 52(3):1130–1140, 2006.
18. L. Trevisan and S. P. Vadhan. Extracting randomness from samplable distributions. In *Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science*, pages 32–42, 2000.
19. S. Vadhan. Pseudorandomness (draft survey). http://people.seas.harvard.edu/ salil/pseudorandomness/, 2012.

20. B. Waters. Efficient identity-based encryption without random oracles. In *Advances in Cryptology – EUROCRYPT '05*, pages 114–127, 2005.
21. H. Wee. Dual projective hashing and its applications - lossy trapdoor functions and more. In *Advances in Cryptology – EUROCRYPT '12*, pages 246–262, 2012.
22. D. Wichs. Barriers in cryptography with weak, correlated and leaky sources. In *Proceedings of the 4th Innovations in Theoretical Computer Science Conference*, 2013.